VisualAge Pacbase 2.5

DMSII DBD
REFERENCE MANUAL

DDDM2000021A

Note

Before using this document, read the general information under "Notices" on the next page.

According to your license agreement, you may consult or download the complete up-to-date collection of the VisualAge Pacbase documentation from the VisualAge Pacbase Support Center at:
http://www.software.ibm.com/ad/vapacbase/support.htm

Consult the Catalog section in the Documentation home page to make sure you have the most recent edition of this document.

First Edition (November 1993)

This edition applies to the following licensed programs:

- VisualAge Pacbase Version 2.0
- VisualAge Pacbase Version 2.5

Comments on publications (including document reference number) should be sent electronically through the Support Center Web site at:
http://www.software.ibm.com/ad/vapacbase/support.htm
or to the following postal address:
IBM Paris Laboratory
VisualAge Pacbase Support
30, rue du Château des Rentiers
75640 PARIS Cedex 13
FRANCE

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

NOTICES

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Intellectual Property and Licensing
International Business Machines Corporation
North Castle Drive, Armonk, New-York 10504-1785
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of information which has been exchanged, should contact:

IBM Paris Laboratory
SMC Department
30, rue du Château des Rentiers
75640 PARIS Cedex 13
FRANCE

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

IBM may change this publication, the product described herein, or both.

TRADEMARKS

IBM is a trademark of International Business Machines Corporation, Inc.
AIX, AS/400, CICS, CICS/MVS, CICS/VSE, COBOL/2, DB2, IMS, MQSeries, OS/2, PACBASE, RACF, RS/6000, SQL/DS, TeamConnection, and VisualAge are trademarks of International Business Machines Corporation, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.
UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

All other company, product, and service names may be trademarks of their respective owners.

TABLE OF CONTENTS

1. INTRODUCTION 7

1.1. PACBASE-PACLAN-PACLAN/X FUNCTIONS 8
1.2. PRESENTATION OF D.B.D. FUNCTION 11
1.3. PRINCIPLES OF DESCRIPTION 12

2. USE OF THE FUNCTION WITH DMSII 13

2.1. INTRODUCTION 14
2.2. PRINCIPLE OF UTILIZATION 16

3. "DATA ITEM" = DATA ELEMENT 17

3.1. DEFINITION OF AN "ITEM" (E.....) 18

4. "DATASET" = SEGMENT 27

4.1. DEFINITION OF A "DATASET" (S....) 28
4.2. DESCRIPTION OF A "DATASET" (S....CE) 33

5. DATABASE = BLOCK 42

5.1. DEFINITION OF A DATABASE (B.....) 43
5.2. DESCRIPTION OF A DATABASE (B.....DC) 49
5.3. GENERAL DOCUMENTATION (-G ET -DCNNNG) 55

6. ACCESS MODES 62

6.1. ON-LINE 63
6.2. BATCH 73
6.3. GENERATION AND/OR PRINTING 76

7. EXAMPLE 77

7.1. PRESENTATION 78
7.2. SCREENS / BATCH TRANSACTIONS 82
7.3. DESCRIPTION OF DASDL GENERATED ELEMENTS 84

1. INTRODUCTION

PAGE 8
INTRODUCTION 1
PACBASE-PACLAN-PACLAN/X FUNCTIONS 1

1.1. PACBASE-PACLAN-PACLAN/X FUNCTIONS

THE VisualAge Pacbase Application Development Solution

VisualAge Pacbase is an Application Development tool operating on mainframe, OS/2, UNIX or Windows NT. It has been designed to ensure the complete management of various information systems.

Consistency is ensured by all the data being stored in one Specification database and managed in a unique way by the System.

INTRODUCTION 1
PACBASE-PACLAN-PACLAN/X FUNCTIONS 1

VISUALAGE PACBASE PRODUCTS

VisualAge Pacbase is a modular AD solution which is composed of two main products - Pacdesign for application design, Pacbench for application development.

Pacdesign and Pacbench are used to populate the Specifications Database and to ensure the maintenance of existing applications. Each product includes several functions.

Basic Functions

Dictionary
Structured Code
Personalized Documentation Manager (PDM-PDM+)

Generators

On-Line Systems Development
Client/Server Facility
Batch Systems Development
COB / Generator

Database Description

DBD
DBD-SQL

Application Revamping

Pacbench Automatic Windowing (PAW) (releases older than VisualAge Pacbase 2.0)

Pacbase Web Connection

INTRODUCTION 1
PACBASE-PACLAN-PACLAN/X FUNCTIONS 1

Quality Control

Pacbench Quality Control (PQC)
Quality Control Extensibility

Table Management

Pactables

Production Turnover and Follow-up

Production Environment (PEI)
PacTransfer
Development Support Management System (DSMS)
PC function: revamped DSMS (in releases older than VisualAge Pacbase 2.0)

Additionnal services
Pac/Impact
Dictionary Extensibility
Pacbase Access Facility (PAF-PAF+)
DSMS Access Facility (DAF)
Methodology (Merise, YSM, etc.)
Sub-networks comparison utilities
Rename/move entity utility (RMEN)
Journal Statistics utility (ACTI)
RACF / TOPSECRET Security Interface
ENDEVOR
VisualAge Smalltalk-VisualAge Pacbase bridge
Team Connection-VisualAge Pacbase bridge

INTRODUCTION 1
PRESENTATION OF D.B.D. FUNCTION 2

1.2. PRESENTATION OF D.B.D. FUNCTION

INTRODUCTION TO THE D.B.D. FUNCTION

The Database Description function automatically generates database descriptions adapted to the database management system in use. This is done by using segment and relationship descriptions defined during the application analysis phase.

The DBD function can generate the description of the following DBMS's:

. Relational databases,

. Network databases (CODASYL),

. Hierarchical databases (DL/1),

. Physical File - AS/400 databases and TANDEM DDL,

. TurboImage databases,

. DMSII databases.

Each one of these DBMS's is documented in a specific Reference Manual.

DBD/RELATIONAL SQL

This function can only be used in conjunction with the Dictionary: data defined in the Specifications Dictionary (whether or not the METHODOLOGY function is being used) can be used to generate database descriptions.

This information is described through a database description language which is independent of the DBMS in use. This allows the user to generate different descriptions from the same source.

INTRODUCTION 1
PRINCIPLES OF DESCRIPTION 3

1.3. PRINCIPLES OF DESCRIPTION

DESCRIPTION PRINCIPLES

In this manual, the entities and screens managed by VisualAge Pacbase are described in two parts:

. An introductory comment explaining the purpose and the general characteristics of the entity or screen,

. A detailed description of each screen, including the input fields for both on-line (screens) and batch (forms) data entry into the Database.

Since input screens and batch forms usually contain the same fields, their descriptions are often identical.

All on-line fields described in this manual are assigned an order number. These numbers are printed in bold italics on the screen examples which appear before the input field descriptions and allow for easy identification of a given field. The numbers are circled on the batch forms.

For certain descriptions, there may be slight differences between the screen and the corresponding batch form. This can be explained by the fact that batch mode is less flexible than on-line mode and often needs additional input fields for some indicators which already exist on the screen.

In addition, the user may find that the field sequence on a screen is different from the field sequence on the corresponding batch form. If that occurs, the numbers referencing the fields may not appear in ascending sequence on either the screen example or the batch form.

>>>> If you use the VisualAge Pacbase WorkStation, the graphical interface of the corresponding windows is described in the VisualAge Pacbase WorkStation Reference Manual.

NOTES: Each type of Database Block has a specific description. However, several Database Block types may use the same Batch Form.

As a result, fields on the Batch Form may have different meanings or may not be used, depending on the type of Database Block.

2. USE OF THE FUNCTION WITH DMSII

2.1. INTRODUCTION

INTRODUCTION

This reference manual describing the DMSII database is not meant to be a training manual of DMSII techniques.

Initial knowledge of both DMSII and the System Specifications Dictionary function is necessary.

This manual contains many examples in order to guide the user during the realisation of a DMSII database in the system Specifications Dictionary.

The Specifications Dictionary function

The objective of the Specifications Dictionary is to manage logical descriptions of the various external views. In order to achieve this objective, it uses the following Sysyem entities:

. Data Elements,

. Segments,

. Database blocks,

. General documentation lines associated to Database blocks.

USE OF THE FUNCTION WITH DMSII 2
INTRODUCTION 1

EQUIVALENT TERMINOLOGY

Note:

A DMSII structure is a set of components, each one with its own syntax. A "dataset" is made of "items".

A Database Block calls Segments, each Segment is a list of Data Elements.

The equivalent terminology is illustrated in the following chart:

DMSII Data	PACBASE EQUIVALENT
Structure	Database Block
Dataset	Segment
Data Item	Data Element
Component : Dataset, Access, Set Subset, Link, Remap	Description Line

First of all, to build a DMSII structure, the called DMSII entities must be defined and described. The description performs the calls of these entities components.

The Data Elements generating DMSII data must initially be defined in the System Specifications Dictionary.

USE OF THE FUNCTION WITH DMSII 2
PRINCIPLE OF UTILIZATION 2

2.2. PRINCIPLE OF UTILIZATION

UTILIZATION PRINCIPLE

A Database Block allows the generation of a DMSII structure. The System D.A.S.D.L. extracts all the information initially entered in the Specifications Dictionary (logical level information).

This information comes from the definition lines, the description lines and the general documentation lines of the Database Blocks.

From a description line, the System can find a Segment description and the Data Elements which belong to to it.

3. "DATA ITEM" = DATA ELEMENT

"DATA ITEM" = DATA ELEMENT 3
DEFINITION OF AN "ITEM" (E.....) 1

3.1. DEFINITION OF AN "ITEM" (E.....)

DEFINITION OF AN "ITEM"

An Item is comparable to a Data Element and can be defined by a Data Element definition line. This line can be accessed with the choice:

CH: E.....

FORMATS

The System generates the data type and length from the internal format.

The types which can be directly interpreted are:

-Alpha : non numerical Display usage (D),

-Numerical : NUMBER usage (N),

-Real : BINARY usage (P).

NOTE: The System usage is indicated between brackets.

All the System formats lead to a generation. An error message appears at the end of the DASDL and points out the non-standard cases. It is the user's responsability to check the compatibility of the result.

The boolean type and boolean field do not correspond to the System formats. However, it is possible to get these types in the Segment description. Then, the DMSII Database can include boolean data.

```
--------------------------------------------------------------------------------
!                                                              *PDMCA.PDEV.HP3.8!
!                                                                               !
! DATA ELEMENT CODE   1  AGE                                                    !
!                                                                               !
! NAME...............:2  AGE                                                    !
! TYPE...............:3  R                                                      !
!                                                                               !
! INPUT FORMAT.......:5  9(2)                               LENGTH...:     2 !
! INTERNAL FORMAT....:6  9(2)                USAGE :7  N    LENGTH...:     2 !
! OUTPUT FORMAT......:8  9(2)                     Z:9       LENGTH...:     2 !
!                                                                               !
! EXPLICIT KEYWORDS..: 10                                                       !
!                                                                               !
! PARENT ELEMENT.....: 11                                                       !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
! SESSION NUMBER.....: 0851       LIBRARY......: HP3    LOCK....:               !
!                                                                               !
!                                                                               !
! O: C1 CH:                            ACTION:                                  !
--------------------------------------------------------------------------------
```

"DATA ITEM" = DATA ELEMENT 3
DEFINITION OF AN "ITEM" (E.....) 1

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	6		DATA ELEMENT CODE (REQUIRED) Enter the mnemonic code which references the data element independently from any data structure, report or screen to which the data element might belong. There is no need to include a report, screen or segment code in the Data Element code since the System does it automatically. This code consists of alphabetic or numeric characters only. Some Data Element codes are reserved by the System for use in data structures, reports or screens and cannot be defined in the Specifications Dictionary:
		SUITE	Prohibited. This code is reserved for the System for program generation.
		FILLER	Data Element that is used for the alignment of fields. Options of the BSD Function: Error Verification fields on transaction files:
		ENPR	Used for Data Element error verification.
		GRPR	Used for Segment error verification.
		ERUT	Used for user defined errors. For more information see DATA ELEMENT CODE on the Segment Call of Elements (-CE) screen. For Reports:
		LIGNE	Reserved for the placement and alignment of the layout line.
		LSKP	Reserved usage only in the '00' Report Structure. See STRUCTURE NUMBER on the Report Call of Elements (-CE) screen.
		SAUT	Reserved usage. This code is the counterpart of LSKP and used with the French version of the System. Options of the OLSD Function:
		ERMSG	Data Element for the placement of the error message.
		LIERR	Reserved usage. This code is the counterpart of ERMSG and used with the French version of the System.
		PFKEY	Used to represent the programmable function keys.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		*PASWD	(IMS only): Used for passwords on a specific screen. The code of the Data Elements provided with the product begins with ".". For the Data Elements you define, you should not use codes beginning with a ".". For more information, see DATA ELEMENT CODE OR SCREEN CODE TO CALL on the On-Line Screen Call of Elements (-CE) screen.
2	36		NAME OF DATA ELEMENT (REQ. IN CREATION) This name should be as explicit as possible. Words used here become implicit keywords (subject to limitations specified in Subchapter "HOW TO BUILD THE THESAURUS", Chapter "KEYWORDS", in the SPECIFICATIONS DICTIONARY Reference Manual). This name appears in documentation and in user manuals and volumes each time the data element is used. It is also possible to list data elements sorted by name. In IMS: Use uppercase.
3	1		TYPE
		P	Property: Elementary piece of information defined at the conceptual level. Note: the FORMAT is optional.
		R	Real Data Element (Default value): elementary piece of information, defined at the Specifications Dictionary level. D.B.D. function: CODASYL elementary data, Relational column.
		A	ALIAS Data Element: This value is used in conjunction with the 'A*' value in the DATA STRUCTURE CODE IN GENER. DESCR. field with the 'DATA' PIA, causes the NAME OF DATA ELEMENT to be generated, rather than the standard element name.
4	1		FORMAT TYPE Batch mode only. This field is used to distinguish which format is being entered in the INPUT, INTERNAL or OUTPUT FORMAT field in batch mode data entry.
		E	Input format.
		I	Internal format (default value).

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		S	Output format. For the input and output formats, only the first ten characters are recognized.
5	10		INPUT FORMAT Not used with the DBD function.
6	10		INTERNAL FORMAT Format normally used in system files (permanent, data-base and temporary files) and in screen input fields. Like the INPUT FORMAT, the INTERNAL FORMAT will be automatically used in the data segment descriptions. For batch programs, the user may select the format type on the Program Call of Data Structures (-CD) screen. It is also used (with the necessary transformations) in screen descriptions (input fields). (Refer to screen description in the ON-LINE SYSTEMS DEVELOPMENT Reference Manual). The internal format must be coded like a COBOL pic-ture (without print characters). The 'INTERNAL USAGE' clause is associated with this format. For data elements that represent a date, it is pos-sible to assign a symbolic format: Display type formats (input):
		D	Without century (DDMMYY or MMDDYY).
		C	With century (DDMMCCYY or MMDDCCYY).
			Internal type formats:
		I	Without century (YYMMDD).
		S	With century (CCYYMMDD).
			Extended type formats (output) (with slashes):
		E	Without century (DD/MM/YY or MM/DD/YY).
		M	With century (DD/MM/CCYY or MM/DD/CCYY).
		G	Gregorian format (CCYY-MM-DD).

"DATA ITEM" = DATA ELEMENT 3
DEFINITION OF AN "ITEM" (E.....) 1

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		T	TIME format (HH:MM:SS).
		TS	TIMESTAMP format METHODOLOGY function: This field may be left blank for a property. For details on the use of the formats with the various types of database blocks, see the summary tables in chapter "COLUMNS: DATA ELEMENTS" of the "Relational SQL Database Description" Reference Manual.
7	1		INTERNAL USAGE Corresponds to the COBOL 'USAGE' clause.
		D	DISPLAY (default option), all hardware. Required for data elements indicating dates.
		C	COMPUTATIONAL (binary), IBM or equivalent; COMPUTATIONAL-4 (binary), IBM SYSTEM 38; COMPUTATIONAL-4 IBM 3-15D, COMPUTATIONAL-6 ICL 2900.
		R	COMPUTATIONAL SYNCHRONIZED RIGHT, IBM or equivalent; This value is preferable to 'C' when binary data are aligned on even addresses, since corresponding COBOL statements are more efficient.
		B	COMPUTATIONAL-1 ICL 1900. BINARY-1 UNISYS 1100 associated with format 1(n).
		S	COMPUTATIONAL SYNCHRONIZED RIGHT ICL 1900.
		N	COMPUTATIONAL-4 aligned on a half-byte. The user must add the complement if the length is uneven.
		P	COMPUTATIONAL-1 BULL 66, 6000 and DPS8.
		L	COMPUTATIONAL-1 SYNCHRONIZED RIGHT ICL 1900.
		Q	COMPUTATIONAL BULL 66, 6000 and DPS8.
		F	COMPUTATIONAL-1 IBM or equivalent. COMPUTATIONAL-9 BULL DPS7. COMPUTATIONAL-11 BULL 66 and DPS8. Relational DBD : floating point, simple precision.
		T	COMPUTATIONAL-3 PACKED SYNC. BULL 66 and DPS8.
		X	DISPLAY SIGN IS TRAILING SEPARATE CHARACTER.
		G	COMPUTATIONAL SYNCHRONIZED RIGHT ICL 2900

"DATA ITEM" = DATA ELEMENT 3
DEFINITION OF AN "ITEM" (E.....) 1

NUM LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		AND COMPUTATIONAL-5 MICROFOCUS.
	7	COMPUTATIONAL-5 ICL 2900.
	K	COMPUTATIONAL CDC. COMPUTATIONAL UNISYS 1100 (COBOL 85)
	M	COMPUTATIONAL-1 CDC.
	N	COMPUTATIONAL UNISYS-A
	O	COMPUTATIONAL-4 UNISYS 1100
	U	COMPUTATIONAL-1 UNISYS 1100.
	W	COMPUTATIONAL-2 UNISYS 1100. COMPUTATIONAL-12 BULL 66 and DPS8. RELATIONAL DBD : floating point, double precision.
	H	COMPUTATIONAL UNISYS 1100. BINARY UNISYS 1100 (COBOL 85)
	8	COMPUTATIONAL BULL 66 COBOL 74 and DPS8.
	9	COMPUTATIONAL-3 BULL 66 COBOL 74 DPS7 and DPS8.
	J	COMPUTATIONAL-6 BULL 66 COBOL 74 DPS7 and DPS8. REAL UNISYS-A.
	Y	DB-KEY BULL 66 DM4 and DPS8. POINTER IBM.
	I	DISPLAY-1 Unisys 1100
	5	COMPUTATIONAL-1 BULL 64 66 MINI-6 COBOL 74 DPS7 DPS8
	6	COMPUTATIONAL-2 BULL 64 66 MINI-6 COBOL 74 DPS7 DPS8
	3	COMPUTATIONAL-3 IBM or equivalent. COMPUTATIONAL BULL 64 MINI-6 DPS7. COMPUTATIONAL-3 (packed decimal) IBM SYSTEM 38. PACKED-DECIMAL UNISYS 1100 (COBOL 85)
	0	COMPUTATIONAL-7 BULL 66 and DPS8.
	1	DISPLAY-1 NCR (signed extended decimal). DISPLAY SIGN LEADING SEPARATE - UNISYS 1100, DPS8, IBM, TANDEM, DPS7.
	4	DISPLAY-2 NCR (unsigned packed decimal).
	2	DISPLAY-2 BULL = DISPLAY, fields are compared in

"DATA ITEM" = DATA ELEMENT 3
DEFINITION OF AN "ITEM" (E.....) 1

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			accordance with the "commercial collating sequence" and not in accordance with the standard BULL sequence.
		Z	In batch mode only: this option, which is only used with an output format, allows for the generation of a 'BLANK WHEN ZERO' clause with the Batch S.D. function. METHODOLOGY function: This field may be left blank for a property.
8	27		OUTPUT FORMAT Not used by the DBD function.
9	1		BLANK WHEN ZERO CLAUSE This field is not used when defining a data element used to generate a CODASYL elementary data element or a relational column.
10	55		EXPLICIT KEYWORDS This field allows the user to enter additional (explicit) keywords. By default, keywords are generated from an occurrence's clear name (implicit keywords). This field only exists on-line. In batch mode, keywords are entered on Batch Form 'G'. Keywords must be separated by at least one space. Keywords have a maximum length of 13 characters which must be alphanumeric. However, '=' and '*' are reserved for special usage, and are therefore not permitted in keywords. Keywords are not case-sensitive: upper-case and lower-case letters are equivalent. NOTE: Characters bearing an accent and special characters can be declared as equivalent to an internal value in order to facilitate occurrence search by keywords. Refer to the Operations Manual - Part II "Administrator's Guide", Chapter "Database Management Utilities", Subchapter "PARM: Update of User Parameters". A maximum of ten explicit keywords can be assigned to one entity. For more details, refer to Chapter "KEYWORDS" Subchapter "BUILDING THE THESAURUS" in the SPECIFICATIONS DICTIONARY Reference Manual.
11	6		PARENT ELEMENT CODE Allows data elements sharing the same characteristics

"DATA ITEM" = DATA ELEMENT 3
DEFINITION OF AN "ITEM" (E.....) 1

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			to be defined under different codes. If a parent data element is indicated, the data element takes on the characteristics of the parent by default. These can be modified at the child level. The parent data element must have been defined previously. METHODOLOGY function: ---------- The notion of 'Parent Data Element' has no significance at the definition level of a property.

4. "DATASET" = SEGMENT

4.1. DEFINITION OF A "DATASET" (S....)

DEFINITION OF A "DATASET"

A "Dataset" is similar to a segment and is defined by a Segment definition line. This line can be accessed with the following choice:

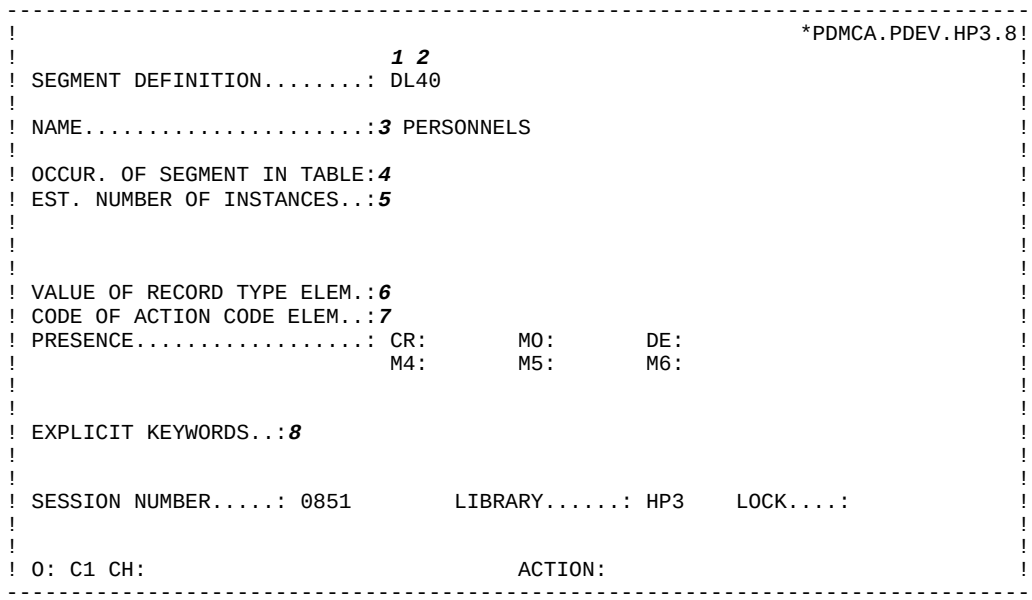
CH: S....

A Segment can be used to generate either a "dataset" or a "remap". The type of generation a Segment will perform is defined when it is called in the Database.

PREREQUISITES

The Data Structure on which the Segment depends must be defined.

"DATASET" = SEGMENT 4
DEFINITION OF A "DATASET" (S....) 1



NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			DATA STRUCTURE / SEGMENT CODE
1	2		DATA STRUCTURE CODE (REQUIRED) This code is made up of two alphanumeric characters. This is a logical code internal to the Database and therefore independent of the names used in Database Blocks and Programs.
2	2		SEGMENT NUMBER (REQUIRED) The first character must be numeric and the second either numeric or alphabetic. However the second character can be alphabetic only if the first character is other than zero.
		00	For standard files: Used to indicate the common part of records in a file, located at the beginning of each record (Default). The control break sort keys, the record type and the keys of indexed files are contained in this Segment. A file does not necessarily have a common part. Records on files with only one type of record should be coded as a '00' Segment. With the Pactables function, this value is not allowed.
		01-99	Designates a specific Segment. The common part Data Elements are automatically concatenated with each specific part Segment. Although a data element may not be used twice in the same Segment, it may be used in both the common part and in one or more specific Segments (except data structures used as Tables).
3	36		SEGMENT CLEAR NAME (REQ. IN CREATION) This name must be as explicit as possible because it is used in the automatic building of keywords, as detailed in chapter "Keywords" in the SPECIFICATIONS DICTIONARY.
4	4		OCCURRENCES OF SEGMENT IN TABLE PURE NUMERIC FIELD WITH THE BATCH SYSTEMS DEVELOPMENT function: This is the amount of space reserved for a Segment in memory (USAGE OF DATA STRUCTURE 'T' or 'X', or RECORD

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			TYPE = 3, or 4. For tables (USAGE OF DATA STRUCTURE 'T' or 'X'), the default value at generation time is 100. Pactables: This field is strictly for documentation purposes. CLIENT/SERVER FACILITY: The value entered in this field indicates the repetitive read or update capacity of the server which calls the Logical View. This capacity is expressed by a maximum number of repetitions. The Logical View can then be used as a repeated structure. NOTE: The use of a Logical View in a card layout does not exclude its use in a row layout. It is therefore strongly recommended to systematically fill in this field. Moreover, the entered value must be high enough to limit the exchanges between the client and the server.
		999	Maximum authorized value.
5	9		ESTIMATED NUMBER OF INSTANCES PURE NUMERIC FIELD For the Batch Systems Development function, this field is used to specify the estimated number of occurrences for a segment in a database or in a standard file. For the METHODOLOGY function, this field is used for activity calculation on the record or set using the Segment (on-line only). For the DBD function, this field is used to specify the application number of an entity in a SOCRATE/CLIO Block.
6	10		CODE / VALUE OF RECORD TYPE ELEMENT For a Relational Table or View, this field is used to specify the external name between quotes. This field is not used to define a CODASYL record.
7	36		CODE OF ACTION CODE ELEMENT This field is not used to define a CODASYL record or a

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			Relational Table or View.
8	55		EXPLICIT KEYWORDS This field allows the user to enter additional (explicit) keywords. By default, keywords are generated from an occurrence's clear name (implicit keywords). This field only exists on-line. In batch mode, keywords are entered on Batch Form 'G'. Keywords must be separated by at least one space. Keywords have a maximum length of 13 characters which must be alphanumeric. However, '=' and '*' are reserved for special usage, and are therefore not permitted in keywords. Keywords are not case-sensitive: upper-case and lower-case letters are equivalent. NOTE: Characters bearing an accent and special characters can be declared as equivalent to an internal value in order to facilitate occurrence search by keywords. Refer to the Operations Manual - Part II "Administrator's Guide", Chapter "Database Management Utilities", Subchapter "PARM: Update of User Parameters". A maximum of ten explicit keywords can be assigned to one entity. For more details, refer to Chapter "KEYWORDS" Subchapter "BUILDING THE THESAURUS" in the SPECIFICATIONS DICTIONARY Reference Manual.

4.2. DESCRIPTION OF A "DATASET" (S....CE)

DESCRIPTION OF A "DATASET"

A "Dataset" is similar to a Segment and is described in the same way as a Segment. The description screen of a Segment is called with the following choice:

CH: S....CE

This description is the list of "items" in the "Dataset" or in the "remap". It is a list of calls of Data Elements in the Segments.

PREREQUISITES

The "dataset" and the called "items" must be defined.

INFORMATION RECOGNIZED

The only data to have an impact on the block generated program are the ones indicated and entered in the -CE.

-Number of repetitions: for OCCURS clause.

-Number of Data Element within a group: for GROUP type.

-Access key or sort indicator: for BOOLEAN types, FIELD, the "remap regrouping" and VIRTUAL. The item RECORD TYPE is also identified here.

-Presence indicator: first position for the REQUIRED clause.

-Update/table: for DEPENDING ON of OCCURS.

Notes:
.A boolean field is indicated with a GROUP and the indicator FIELD.
.In a "remap", a virtual boolean or field is not automatically obtained (the same field is used).

"DATASET" = SEGMENT 4
DESCRIPTION OF A "DATASET" (S...CE) 2

ROLE IN THE GENERATION

A Segment description is used to describe a dataset or a "remap". A "remap" is therefore a Segment in which the Data Elements chosen are called automatically. It is then impossible to use the functionality that automatically hides Data Elements (HIDDEN). A group Data Element in the dataset can be called without this notion.

Note: For the groups in a "remap", the number of repetitions (OCCURS) and the presence indicator (REQUIRED) are ignored.

"DATASET" = SEGMENT 4
DESCRIPTION OF A "DATASET" (S...CE) 2

```
-------------------------------------------------------------------------------
!                          1 2                                *PDMCA.PDEV.HP3.8!
! SEGMENT CALL OF ELEMENTS DL40 PERSONNELS                                      !
! 3 4     5      7          8 9   10 11  12   <      13    >  14        15     !
! A LIN : ELEM.  INT.FORM.  U OCC GR K CMD456 CONT VALUE/SFC  UPD/TRGET DOC LIB!
!   100 : NBPER                                                         085!
!   200 : NOMC                     2                                    085!
!   210 : NOM                                                           085!
!   220 : PRENOM                                                        085!
!   250 : SEXE                       B                                  085!
!   260 : AGE                                                           085!
!   270 : SSNO                         O                                085!
!   300 : DPT                                                           085!
!   310 : RANG                                                          085!
!   320 : SALAIR                                                        085!
!   400 : IDCOUR               8                                        085!
!   500 : TELEPH                                                        085!
!   600 : SUPER                                                         085!
!       :                                                                   !
!       :                                                                   !
!       :                                                                   !
!       :                                                                   !
!       : NAME    : 6                                                       !
! *** END ***                                                               !
! O: C1 CH:                                                                 !
-------------------------------------------------------------------------------
```

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			DATA STRUCTURE / SEGMENT CODE
1	2		DATA STRUCTURE CODE (REQUIRED) This code is made up of two alphanumeric characters. This is a logical code internal to the Database and therefore independent of the names used in Database Blocks and Programs.
2	2		SEGMENT NUMBER (REQUIRED) The first character must be numeric and the second either numeric or alphabetic. However the second character can be alphabetic only if the first character is other than zero.
		00	For standard files: Used to indicate the common part of records in a file, located at the beginning of each record (Default). The control break sort keys, the record type and the keys of indexed files are contained in this Segment. A file does not necessarily have a common part. Records on files with only one type of record should be coded as a '00' Segment. With the Pactables function, this value is not allowed.
		01-99	Designates a specific Segment. The common part Data Elements are automatically concatenated with each specific part Segment. Although a data element may not be used twice in the same Segment, it may be used in both the common part and in one or more specific Segments (except data structures used as Tables).
3	1		ACTION CODE (REQUIRED)
4	3		LINE NUMBER PURE NUMERIC FIELD It is advisable to begin with line number '100' and then number in intervals of 20. This facilitates subsequent line insertions, as necessary.
5	6		DATA ELEMENT CODE ELEMENTARY DATA ELEMENT DEFINED IN THE DICTIONARY --- The Data Element automatically assumes the character-

"DATASET" = SEGMENT 4
DESCRIPTION OF A "DATASET" (S...CE) 2

NUM LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		istics defined at the Specifications Dictionary level. If the Data Element is used as a group, its format depends on the characteristics of the elementary Elements that make up the group. If the group is used as a key (sort or access key), the composite format of the elementary Elements must be compatible with the format specified for the group. DATA ELEMENT NOT DEFINED IN THE DICTIONARY --- The name and/or format of undefined Data Elements must be indicated at the segment level. RESERVED DATA ELEMENT CODES --- Prohibited. This code is reserved for the System for program generation.
	SUITE	
	FILLER	Data Element that is used for the alignment of fields. OPTIONS OF THE BATCH SYSTEMS DEVELOPMENT FUNCTION --- These codes (when used) precede other entries made in this field, in the sequence described below.
	ENPR	Used to store Element error verifications in a transaction file. The length is n + 1 where n = either the total number of elementary Elements in the file, or the number of elementary Elements in the '00' Segment added to the largest non-00 Segment. ("Largest" here means the most elementary Elements.) This depends upon the value entered in the RESERVED ERROR CODES IN TRANS FILE field on the Call of Data Structures (-CD) screen.
	GRPR	Used to store Segment error verifications. Its length is n + 1 where n = the number of records.
	ERUT	Used to store error verifications for users. Normally, these last three Data Elements are used in transaction files for error verification fields. When used in other types of files as "optional" Data Elements, they may be used as group fields whose generation may be invoked or suppressed according to the option selected in the RESERVED ERROR CODES IN TRANS.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			FILE field. (Note: this will affect the elementary Elements within the group as well.) CALLING DATA AGGREGATES ---------- A SEGMENT CODE or a Model Entity code (Relationship or Object in the METHODOLOGY function) can be entered in this field. The called data aggregate will be interpreted as if the individual Elements that make it up had been entered. The NO. OF ELEMENTARY ELEMENTS IN GROUP field is used to identify data aggregate calls. Enter the code at the location the elements are to be included in the Segment description. In O:C2, the level of 'nesting' is displayed in the Action Code (up to four levels). The number of authorized nesting levels varies according to the type of generator. Up to 4 nesting levels are authorized for data generation and PAF use. CONTINUATION LINES ---------- It is possible to create continuation lines. This may be necessary if there are many validations on a Data Element. In this case, leave the DATA ELEMENT CODE field blank, and use a LINE NUMBER value that sequentially follows that of the line where the Data Element code was entered.
6	18		NAME OF DATA ELEMENT It is required for a Data Element which is not defined in the Specifications Dictionary. However, it is optional for a data aggregate or a FILLER. Note: For on-line entry of Data Elements that are not declared in the Dictionary, this field cannot be used to input more than one Data Element at a time. There is actually only one available field on this screen, whether for input or for display. To define an Element at the Segment level : - Enter the Element code (and possibly the format)

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			on the -CE, line nnn, - On the 'name' line, repeat the line number (nnn), and indicate the name (18 characters maximum), - Use the C2 option to view the name and format. Note: If several undefined Elements have been named in this fashion, the name displayed will be the one that refers to the Element with the lowest line number on the display. To view a specific Element's name use the CHOICE field, selecting the appropriate Element by line number. Example: O: C2 CH: -ce130 will display all Data Elements starting with the one on line 130. If it is an undefined Element, its name will appear in the NAME OF DATA ELEMENT field.
7	10		DATA ELEMENT INTERNAL FORMAT It is required only in the following cases : - For an elementary Data Element not defined in the Dictionary (COBOL format), - For a group Data Element that is or belongs to a key; its length must be the sum of the lengths of its elementary Data Elements, - For a FILLER-type field. It is the internal format; input and output formats will be the same (but with usage Display). It is defined as on a Data Element Definition screen.
8	1		INTERNAL USE For Data Elements not defined in the Specifications Dictionary when the INTERNAL FORMAT OF DATA ELEMENT field has been given a value, enter the appropriate USAGE (default : 'D' for DISPLAY). For valid values, see the USAGE field on the Data Element Definition Screen.
9	3		OCCURRENCES (COBOL "OCCURS" CLAUSE) PURE NUMERIC FIELD This field represents the 'OCCURS' clause at an elementary Data Element level, or at a group level (Maximum of 3 levels).

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			It can be changed into an 'OCCURS DEPENDING ON' clause by entering '**' in the UPDATE TARGET field, followed by the counter's Segment and Data Element codes. The COBOL restrictions on the OCCURS clause apply.
10	2		NO. OF ELEMENTARY ELEMENTS IN GROUP PSEUDO NUMERIC FIELD
		1 to 99	For group Data Elements, enter the number of elementary Elements that belong to the group (A Segment call is considered as an elementary Data Element). Groups may contain up to 99 elementary Elements. Group Elements may contain embedded groups however the total number of elementary Elements cannot exceed 99. (The group Data Element codes are not counted). The maximum number of levels of 'nesting' is 9. This field is also used to identify the entity called in the DATA ELEMENT CODE field as Methodology entities or previously defined Segments.
		*M	Call of an Object or a Relationship.
		**	Call of a Segment. SQL DBD function:
		**	Call of a Segment into a view.
11	1		ACCESS KEY OR SORT KEY It allows to manage with DMSII information that PACBASE entities do not know and to get a generation that includes these particularities. "Item" type not defined in the Data Element:
		B	.Boolean
		F	.Field : integer or boolean field
		T	.Record Type : identifier of the variable part For the "remap":
		V	.virtual item
		R	.group defined in the "remap" (It is the case for "remap regrouping")
12	6		DATA ELEMENTS PRESENCE CONTROL

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			Only the field first position is used to indicate the presence of an "item" in a "dataset".
		Blank	Optional "item" (default value).
		O	"Item" REQUIRED.
			This clause will be generated after the "item" definition in the "dataset".
13	14		CONT AND VALUE/SFC FIELDS These fields are not used.
14	16		INDICATION OF OCCURS DEPENDING ON -'UPD/TRGET' FIELD
		**	This means that the "occurs" is "DEPENDING ON" for the Data Element in this line.
15	1		DOCUMENTATION INDICATOR This field is a display field used on-line only. It does not accept input.
		*	General documentation exists for the element on this line. Access to line nnn: -CEnnn Access to the documentation of line nnn: -CEnnnG For more details, see the "GENERAL DOCUMENTATION" chapter in the SPECIFICATIONS DICTIONARY Reference Manual.

5. DATABASE = BLOCK

DATABASE = BLOCK 5
DEFINITION OF A DATABASE (B.....) 1

5.1. DEFINITION OF A DATABASE (B.....)

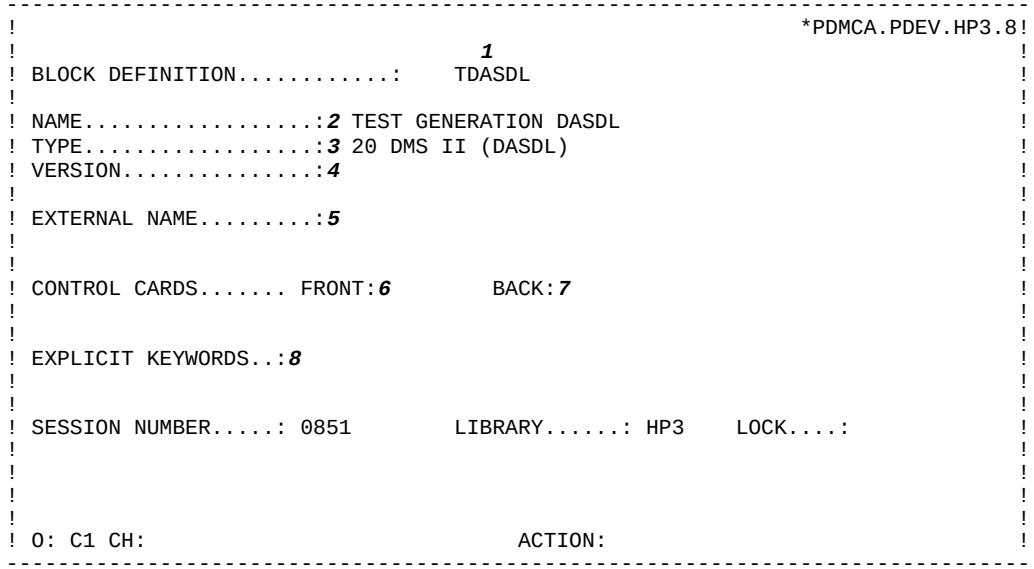
DEFINITION OF A DATABASE

A DMSII Database is similar to a System Block and is defined by a Database Block definition line.

A Database Block is defined with a code, a name and a type.

A Database Block used to generate a DMSII structure is to be defined by type "20".

DATABASE = BLOCK 5
DEFINITION OF A DATABASE (B.....) 1



DATABASE = BLOCK 5
DEFINITION OF A DATABASE (B.....) 1

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	6		BLOCK CODE (REQUIRED) One to six alphanumeric characters.
2	36		NAME OF THE BLOCK (REQ. IN CREATION) This clear name should be as explicit as possible. Words used here become implicit keywords (subject to limitations specified in Subchapter "HOW TO BUILD THE THESAURUS", Chapter "KEYWORDS" in the SPECIFICATIONS DICTIONARY Reference Manual).
3	2		TYPE OF BLOCK (REQ. IN CREATION) For hierarchical or network databases, it is not necessary, when creating a database block, to enter the definitive block type. The selection of a network or hierarchical structure is sufficient at this point. A specific "physical" type must be entered when generating the Data Description Language (DDL).
		TR	Tree-like structure (hierarchical block).
		SE	Group of sets (network block).
			HIERARCHICAL DATABASES - IMS/DL1 ----------
		DP	Physical Database Description.
		DR	Physical Database Description (same as 'DP', but only the data elements referenced as access keys in the segment description are generated in the 'FIELD.....' statements).
		DL	Logical Database Description.
		PC	PCB.
		IP	Primary Index.
		IS	Secondary Index.
		PS	PSB (Assigned at creation. Cannot be modified at a later stage).
			RELATIONAL DATABASES ----------
		Q2	DB2 SQL
		Q3	SQL SERVER
		Q4	DB2/400
		QA	ALLBASE/SQL
		QB	DB2/2 and DB2/6000
		QC	DATACOM/DB
		QG	INGRES/SQL
		QI	INFORMIX-ESQL
		QN	NONSTOP SQL

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		QO	ORACLE (releases earlier than V6)
		QP	ORACLE (from release V6 on)
		QR	RDMS
		QS	SQL/DS
		QT	INTEREL RDBC
		QU	INTEREL RFM
		QV	VAX SQL
		QY	SYBASE
		DB	DB2 (It is recommended to use the Q2 type)
			NETWORK DATABASES ----------
			.CODASYL-DM4 (BULL 66 or DPS8): ----------
		M1	DDL schema, only elementary fields are generated,
		M4	DDL schema, only group fields are generated,
		M2	DMCL schema,
		M3	Sub-schema.
			.CODASYL-IDS2 (BULL 64 or DPS7): ----------
		I1	DDL schema,
		I2	DMCL schema,
		I3	SDDL sub-schema.
			.CODASYL-IDMS: ----------
		D0	DDL schema (Release 10.0),
		D1	DDL schema,
		D2	DMCL schema,
		D3	Sub-schema,
		D4	Sub-schema (Release 5.7).
			.CODASYL-DMS (UNISYS 1100): ----------
		S1	DDL Schema,
		S3	Sub-schema.
			DDL TANDEM ----------
		TD	TANDEM
			AS/400 PHYSICAL FILE ----------

DATABASE = BLOCK 5
DEFINITION OF A DATABASE (B.....) 1

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		PF	AS/400 Physical file (IBM SYS. 38)
		LF	AS/400 Logical file (IBM SYS. 38).
			DDL TURBOIMAGE ----------
		TI	TurboImage Database.
			DMSII DATABASE ----------
		20	DMSII Database (DASDL)
4	4		VERSION This field is not used.
5	8		DATABASE BLOCK EXTERNAL NAME Necessary at generation time. This is the physical name of the System-generated DDL (Data Description Language) module. To obtain a list of blocks sorted by this external name, enter 'LEB' in the CHOICE field. For TurboImage, only the first six characters are processed.
6	1		CONTROL CARDS IN FRONT OF BLOCK Necessary at generation time. Enter the one-character code that identifies the job control card to be inserted before the generated block.
7	1		CONTROL CARDS IN BACK OF BLOCK Necessary at generation time. Enter the one-character code that identifies the job control card to be inserted after the generated block.
8	55		EXPLICIT KEYWORDS This field allows the user to enter additional (explicit) keywords. By default, keywords are generated from an occurrence's clear name (implicit keywords). This field only exists on-line. In batch mode, keywords are entered on Batch Form 'G'. Keywords must be separated by at least one space. Keywords have a maximum length of 13 characters which

NUM LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		must be alphanumeric. However, '=' and '*' are reserved for special usage, and are therefore not permitted in keywords. Keywords are not case-sensitive: upper-case and lower-case letters are equivalent. NOTE: Characters bearing an accent and special characters can be declared as equivalent to an internal value in order to facilitate occurrence search by keywords. Refer to the Operations Manual - Part II "Administrator's Guide", Chapter "Database Management Utilities", Subchapter "PARM: Update of User Parameters". A maximum of ten explicit keywords can be assigned to one entity. For more details, refer to Chapter "KEYWORDS" Subchapter "BUILDING THE THESAURUS" in the SPECIFICATIONS DICTIONARY Reference Manual.

5.2. DESCRIPTION OF A DATABASE (B.....DC)

DESCRIPTION OF A DMSII DATABASE

A DMSII Database is similar to a System Block and is described by a Database Block description line.

CH: B.....DC

This description is a list of elements within the Database.

Six line types are taken into account:

```
. Dataset        ---> 1

. Access         ---> 2

. Set            ---> 3

. Subset         ---> 4

. Link           ---> 5

. Remap          ---> 6
```

DESCRIPTION ORDER

The description lines are ordered by their number. This order corresponds to the generation presentation order. The notion of parent Segment allows interlockings management but does not interfere in the location of generated elements. Attention must be payed to embedded elements (EMBEDDED) especially "accesses", sets and subsets in order to get a good generation.

LINKS

```
Links are not identified by a type but by the line codification.In
the order of the line, the following elements are found:
.Counted          : datell , ffssp , ffss , C
.Self-correction  : datell , C     , ffss , set
.Symbolic         : datell , S     , ffss , set
.Unprotected      : datell , ffssp , ffss , N
.Verified         : datell , ffssp , ffss , datelp

Generated elements are in the ffss dataset:
.ffss-datell IS IN ffssp COUNTED;
.ffss-datell IS IN set;
.ffss-datell IS KEY OF set;
.ffss-datell IS IN ffssp WITH NO PROTECTION;
.ffss-datell IS IN ffssp VERIFIED ON ffssp-datelp;
```

VARIABLE STRUCTURES

In a Dataset there is a variable part and a fixed part.
The fixed part is indicated by a number present in the corresponding field. The item "RECORD TYPE" must be coded in the fixed part and is marked with the letter "T" in the sort key.
Each variable is identifiable by the letter "V" located in the "set or Data Element code" field. The number indicated in the corresponding column is the DMSII internal identifier of the variable record.

```
------------------------------------------------------------------------------
!                                1                            *PDMCA.PDEV.HP3.8!
! DESCRIPTION OF DASDL DMSII  TDASDL TEST GENERATION DASDL                     !
! 2 3     4 5      6    7    8          9   10                                 !
! A LIN : T SET NA DATASET   OPTION     PT. COMMENT                            !
!       :   LIN IT EMB.      SET/IT     N K                                    !
!   100 : 1             DL10 S      *       MAIN FILE : COURSES                !
!   101 : 5 PROF   DL40 DL10 C                                                 !
!   200 : 1        DL10 DL20 U      *       BOOKS                              !
!   210 : 4 LIVK   DL10 DL20 LI     *                                          !
!   300 : 1        DL10 DL30 S      *       STUDENTS                           !
!   301 : 5 SSNO   C    DL30 MFSSET                                            !
!   302 : 5 SSNO1  S    DL30 MFSSET                                            !
!   310 : 3 ETUSET DL10 DL30 IS     * 00002                                    !
!   350 : 3 COUSET      DL10 IS     *                                          !
!   400 : 1             DL40 S      *       PERSONNEL                          !
!   401 : 5 IDCOUR DL10 DL40 C                                                 !
!   402 : 5 TELEPH DL80 DL40 TELEPH                                            !
!   403 : 5 SUPER  DL40 DL40 N                                                 !
!   410 : 3 SS-U-P      DL40 IS     *                                          !
!   420 : 3 U-P-ST      DL40 IS     *                                          !
!   500 : 1             DL50 S      *       REGISTERED                         !
!   501 : 5 TELEPH DL80 DL50 TELEPH                                            !
!                                                                              !
! O: C1 CH:                                                                    !
------------------------------------------------------------------------------
```

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
1	6		BLOCK CODE (REQUIRED) One to six alphanumeric characters.
2	1		ACTION CODE
3	3		LINE NUMBER PURE NUMERIC FIELD It is advisable to begin with line number '100' and then number in intervals of 20. This facilitates subsequent line insertions, as necessary.
4	1		LINE TYPE (REQUIRED) It identifies the DMSII element which is to be defined.
		1	Dataset
		2	Access
		3	Set
		4	Subset
		5	Link
		6	Remap
5	6		SET OR DATA ELEMENT CODE This field has three different meanings:
		.1 and 6 Blank	.For dataset and remap: The segment is comparable to a dataset or a dataset "remap".
		V	The segment describes a variable part.
		.2, 3, 4	.For accesses, sets and subsets: This element name (access, set or subset).
		.5	.For a link: The Data Element code which is a link.
6	4		PARENT SEGMENT CODE Indicates if the affected element is embedded.
		.5	For a link:
		ffss	Indicates the segment which is the link reference for "counted, unprotected, verify link". Differenciates the link type:

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
		C	Self-correction link.
		S	Symbolic link.
7	4		SEGMENT CODE (REQUIRED) DMSII element or reference to this element.
8	6		DATASET, SET/DATA ELEMENT TYPE The meaning changes depending on the line type.
		.1	For the dataset, the dataset type:
		C	-Compact.
		D	-Direct.
		O	-Ordered.
		R	-Random.
		RE	-Restart.
		S	-Standard.
		U	-Unordered.
		.3 et 4	For sets and subsets, set or subset type:
		BV	-Vector Bit.
		IR	-Random Index.
		IS	-Sequential Index.
		LI	-Unordered List.
		OL	-Ordered List.
		Other	-Reference set for the subset.
		.5	For links, three possible contents:
			-Type Distinction
		C	.Counted link.
		N	.Unprotected link.
			-Link reference set
			.Self-correction. .Symbolic link.

NUM	LEN	CLASS VALUE	DESCRIPTION OF FIELDS AND FILLING MODE
			-Data Element code in the reference Segment .Verified link. .6 For a "Remap" : =ffss The equal sign followed by the remapped Segment code.
9	5	NUMER.	NUMBER IDENTIFYING VARIABLE PART This field has different meanings depending on the line type. .1 and 6 For datasets and remaps: The variable part identifier or the "record type" maximum value. .2, 3, 4 For accesses, sets and subsets: Number of items part of the key. .5 For a link : Its number of repetitions (OCCURS).
10	36		COMMENT Associated to the DMSII element.

5.3. GENERAL DOCUMENTATION (-G et -DCnnnG)

ADDITIONAL ELEMENTS OF DESCRIPTION

The definition and description lines of a Database Block provide all the logical information the System needs to generate the source program of the Block. The physical information must not be indicated on these lines but must be typed on the documentation lines corresponding to the Block description lines.

CH: -G (for the Block)
CH: -DCnnnG (for a Block element)

With the general documentation lines, the user can insert comments, commands, descriptions, labels, print requests wherever he wishes to in the generated structure. He can also erase and replace the description the System generates automatically.

Four types of lines are available for the user to insert. Two types of insertions are possible:

-within an element definition lines
-within a particular item of the element

All the lines concerning the entity definition must be at the beginning, all the lines concerning an "item" must be consecutive.

Types of lines :

'V': lines generated before the automatically generated elements.

'P': lines generated between the element automatically generated parts and its description.

'Z': lines generated after the automatically generated elements.

'G': line generated instead of the automatically generated elements.

Item :

The Data Element code is indicated between the 'less than' and the 'greater than' symbols on a general documentation line of the affected entity.

<datel >

This notation is taken into account in the Data Element utilization definition.

In a dataset or a "remap" description, the Data Element marked this way does not interfere in the generation and must not have any line type.
For "access", set and subset, Data Elements marked this way are taken into account for the generation and they must have a type.

Data indication

Most often, this is performed from column two. If the data description starts in column one, a semi-colon line is inserted before the line generated for this data.

DATABASE = BLOCK **5**
GENERAL DOCUMENTATION (-G et -DCnnnG) **3**

```
--------------------------------------------------------------------------------
!                                                                  *PDMCA.PDEV.HP3.8!
! BLOCK DESC GENERAL DOC.         TDASDL TEST GENERATION DASDL                1!
!                                                                               !
! A LIN : T COMMENT                                                       LIB !
!   100 : Z  % FIN DE LA DESCRIPTION DE L'ENREGISTREMENT COURS            0851!
!   110 : Z  POPULATION = 1000                                            0851!
!   150 : Z  VERIFY (DL10-NBHEU GTR 0 AND DL10-TCLASS LEQ 60)             0851!
!   151 : Z         AND DL10-NOPROF NEQ 0                                 0851!
!   500 :   <IDCOUR>                                                      0874!
!   501 : P  REQUIRED                                                     0851!
!   510 :   <NBINS >                                                      0851!
!   511 : G  DL10-NBINS COUNT (300);                                      0851!
!   520 :   <SALLE >                                                      0851!
!   521 : P  NULL IS "NO";                                                0851!
!       :                                                                     !
!       :                                                                     !
!       :                                                                     !
!       :                                                                     !
!       :                                                                     !
!       :                                                                     !
!       :                                                                     !
!       :                                                                     !
!                                                                             !
! O: C1 CH:                                                                   !
--------------------------------------------------------------------------------
```

UTILIZATION

For the Block description (-G)

The four line types are possible.

.G / The elements automatically generated are overwritten.

.V / To indicate generation-print requests (ex:$SET ..).

.P / To enter specification parameters for instance and other Database elements.

.Z / To enter logical descriptions or any data that must be located at the end of a description.

For a description line (-DCnnnG)

The four line types are possible.

.G / The automatically generated elements are overwritten.

.V / To indicate data before the dataset.

.P / To type lines before the dataset description (POPULATION for instance).

.Z / To enter clauses located after the dataset description (Physical options for instance).

For an "item"

Three line types only. The notion of end (Z) is reserved to the Block and the description line.

.V / Before the generated elements concerning the item.

.G / Overwrites the generated elements concerning the item.

This utilization is useful for the codification of a virtual item with a "field" type.

For a group Data Element, the entire group is overwritten.

.P / After generated elements concerning the item.

This type of item will certainly be the most used as it completes an item generation with particular DMSII clauses.

When such a type of line is used, it is the user's responsability to indicate the end of instructions (semi-colon).

Details concerning the "access", set and subset lines.

The marked Data Element can be used to do the following:

.P / codify key items.

.Z / indicate "DATA" items.

This entry gives indication about the Data Element utilization. The remaining space on the line is available to enter additional information, in particular the punctuation.

Example : Generation of a KEY clause for a set on a ffss dataset:

```
P <datel1>,
P <datel2>DESCENDING,
P <datel3>)
generates
KEY IS (
ffss-datel1 ,
ffss-datel2 DESCENDING,
ffss-datel3 )
```

PRESENTATION OF GENERATED ELEMENTS

For the entire block

V : -- line -G
G : INITIALIZE;
P : -- line -G
/ Generated elements from the Block description.
Z : -- line -G

For a description line

.Dataset
V : -- line -DCnnnG
G : DATASET type name
G : "comment"
P : -- line -DCnnnG
G : (
/ Generated elements from the dataset description.
G :)
Z : -- line -DCnnnG
G : ;

.Access
V : -- line -DCnnnG
G : "comment" name
G : ACCESS TO dataset
G : KEY IS (
P : -- line -DCnnnG
Z : -- line -DCnnnG
G : ;

.Set
V : -- line -DCnnnG
G : "comment" name
G : SET OF dataset
G : KEY IS (
P : -- line -DCnnnG
G : type
Z : -- line -DCnnnG
G : ;

For a Data Element

.Elementary Data Element
V : -- line -DCnnnG
G : ffss-datel type (;)
G : REQUIRED (;)
G : OCCURS n (;)
G : DEPENDING ON ffss-datel (;)
P : -- line -DCnnnG

.Group Data Element
V : -- line -DCnnnG
G : ffss-datel type (;)
G : REQUIRED (;)
G : OCCURS n (;)
G : DEPENDING ON ffss-datel (;)
P : -- line -DCnnnG
G : (
G :);

NOTE: A G-type line overwrites the entire group of automatically generated lines and is located on the first line. In this case, the user must enter the right punctuation (semi-colon and brackets) using P-type or Z-type lines.

6. ACCESS MODES

6.1. ON-LINE

DATA ELEMENTS: ON-LINE ACCESS

LIST OF DATA ELEMENTS

CHOICE	SCREEN	UPD
LCEaaaaaa	List of Elements by Code (starting with data element 'aaaaaa').	NO
LNEaaaaaaaaaaaaa	List of Data Elements sorted by name (starting with name 'aaaaaaaaaaaaa'). The sort is performed on the following elements: - the first twenty characters of the clear name, - the code of the Data Element. Note: Child Data Elements with no clear name do not appear on the list	NO
LACEaaaaaaaaaaaaaaaaa	List of Elements by COBOL name (starting with data element 'aaaaaaaaaaaaaaaaaa') For elements from REVERSE ENG.	NO
LALEaaaaaaaaaaaaa	List of data elements sorted by name (starting with name 'aaaaaaaaaaaaa'). Equivalent of 'LNE'.	NO
LAREaaaaaaaaaaaaaaaaaa	List of data elements sorted by relational name (starting with 'aaaaaaaaaaaaaaaaaa').	NO
LFEaaaaaa	List of undefined data elements by code (starting with element 'aaaaaa').	NO
LUEaaaaaa	List of data elements for update (starting with element 'aaaaaa').	YES

DESCRIPTION OF DATA ELEMENT 'aaaaaa'

CHOICE	SCREEN	UPD
Eaaaaaa	Definition of data element 'aaaaaa'.	YES
EaaaaaaDbbb	Description of data element 'aaaaaa' (starting with line number 'bbb').	YES
EaaaaaaGbbb	General Documentation for data element 'aaaaaa' (starting with line number 'bbb').	YES
EaaaaaaATbbbbbb	Text assigned to the data element 'aaaaaa' (starting with text 'bbbbbb').	NO
EaaaaaaX	X-references of data element 'aaaaaa' to all entities.	NO
EaaaaaaXTbbbbbb	X-references of data element 'aaaaaa' to texts (starting with text 'bbbbbb').	NO
EaaaaaaXMbbbbbb	X-references of data element 'aaaaaa' to the Method Entities (starting with Method Entity 'bbbbbb').	NO
EaaaaaaXQbbbbbb	List of entities linked to data element 'aaaaaa' through user-defined relationship 'bbbbbb'.	NO
EaaaaaaXBbbbbbb	X-references of data element 'aaaaaa' to blocks (starting with block 'bbbbbb').	NO
EaaaaaaXBbbbbbbDCddd	X-references of data element 'aaaaaa' to CODASYL-type blocks (starting with block 'bbbbbb', line number 'ddd')	NO

ACCESS MODES **6**
ON-LINE **1**

EaaaaaaXBbbbbbbDHddd	X-references of data element 'aaaaaa' to Hierarchical-type block (starting with block 'bbbbbb', line number 'ddd')	NO
EaaaaaaXBbbbbbbDRddd	X-references of data element 'aaaaaa' to Relational-type block (starting with block 'bbbbbb', line number 'ddd')	NO
EaaaaaaXVbbbbbb	X-references of data element 'aaaaaa' to volumes (starting with volume 'bbbbbb').	NO
EaaaaaaXObbbbbb	X-references of data element 'aaaaaa' to screens (starting with screen 'bbbbbb').	NO
EaaaaaaXObbbbbbWccddd	X-references of data element 'aaaaaa' to work areas (-W) of screen 'bbbbbb' (starting with work area 'cc', line number 'ddd').	NO
EaaaaaaXObbbbbbBccddeee	X-references of data element 'aaaaaa' to Beginning Insertions (-B) of screen 'bbbbbb' (starting with section 'cc', paragraph 'dd', line number 'eee').	NO
EaaaaaaXObbbbbbCPcccccc	X-references of data element 'aaaaaa' to Call of P.M.S.(-CP) of screen 'bbbbbb' (starting with macro-structure 'cccccc').	NO
EaaaaaaXObbbbbbPccddeee	X-references of data element 'aaaaaa' to procedural code (-P) of screen 'bbbbbb' (starting with function/subfunction 'ccdd', line number ' eee').	NO
EaaaaaaXKbbbb	X-references of data element 'aaaaaa' to the key of relational /SQL database blocks (starting with segment 'bbbb').	NO
EaaaaaaXSbbbb	X-references of data element 'aaaaaa' to segments (starting with segment 'bbbb').	NO

ACCESS MODES 6
ON-LINE 1

EaaaaaaXRbbb	X-references of data element 'aaaaaa' to reports (starting with report 'bbb').	NO
EaaaaaaXRbbbCE	X-references of data element 'aaaaaa' to report call of elements (starting with report 'bbb').	NO
EaaaaaaXPbbbbbb	X-references of data element 'aaaaaa' to programs (starting with program 'bbbbbb').	NO
EaaaaaaXPbbbbbbBccddeee	X-references of data element 'aaaaaa' to Begininning Insertions (-B) of program 'bbbbbb' (starting with section 'cc', paragraph 'dd', line number 'eee').	NO
EaaaaaaXPbbbbbbCPcccccc	X-references of data element 'aaaaaa' to Call of P.M.S. (-CP) of program 'bbbbbb' (starting with macro-structure 'cccccc').	NO
EaaaaaaXPbbbbbbSCfusfnnn	X-references of data element 'aaaaaa' to source code (-SC) of 'reversed' program 'bbbbbb' (starting with function/subfunction 'fusf', line number 'nnn')	NO
EaaaaaaXPbbbbbbWccddd	X-references of data element 'aaaaaa' to work areas (-W) of program 'bbbbbb' (starting with work area 'cc', line number 'ddd')	NO
EaaaaaaXPbbbbbbPfusfnnn	X-references of data element to procedural code (-P) of program 'bbbbbb' (starting with function/ subfunction 'fusf', line number 'nnn').	NO
EaaaaaaXPbbbbbb9cccccc	X-references of data element to Pure COBOL Source Code (-9) of program 'bbbbbb' (starting with -9 line 'cccccc').	NO
EaaaaaaXFbbbbbb	X-references of data element 'aaaaaa' to User Entities (starting with UE 'bbbbbb').	NO

ACCESS MODES 6
ON-LINE 1

NOTE: After the first choice of type 'Eaaaaaa', 'Eaaaaaa' can be replaced with '-'.

All notations between parentheses are optional.

SEGMENTS: ON-LINE ACCESS

LIST OF SEGMENTS

CHOICE	SCREEN	UPD
LCSaaaa	List of segments by code (starting with segment 'aaaa').	NO

DESCRIPTION OF SEGMENT 'aaaa'

CHOICE	SCREEN	UPD
Saaaa	Definition of segment 'aaaa'.	YES
SaaaaGbbb	General documentation for segment 'aaaa' (starting with line number 'bbb').	YES
SaaaaATbbbbbb	Text assigned to segment 'aaaa' (starting with text 'bbbbbb').	NO
SaaaaLSPbbbb	List of parent segments for segment 'aaaa' (starting with parent segment 'bbbb').	NO
SaaaaLSCbbbb	List of child segments for segment 'aaaa' (starting with child segment 'bbbb').	NO
SaaaaX	X-references of segment 'aaaa'.	NO
SaaaaXSbbbb	X-references of segment 'aaaa' to segments (starting with segment 'bbbb').	NO
SaaaaXBbbbbbb	X-references of segment 'aaaa' to blocks (starting with block 'bbbbbb').	NO
SaaaaXQbbbbbb	List of entities linked to segment 'aaaa' through user-defined relationship 'bbbbbb'.	NO

SaaaaXVbbbbbb	X-references of segment 'aaaa' to volumes starting with the 'bbbbbb' volume.	NO
SaaaaXPbbbbbb	X-references of segment 'aaaa' to programs (starting with program 'bbbbbb').	NO
SaaaaXPbbbbbbCPcccccc	X-references of segment 'aaaa' to Call of P.M.S. (-CP) of program 'bbbbbb' starting with macro-structure 'cccccc').	NO
SaaaaXPbbbbbbWccddd	X-references of segment 'aaaa' to work areas (-W) of program 'bbbbbb' (starting with work area 'cc', line number 'ddd').	NO
SaaaaXObbbbbb	X-references of segment 'aaaa' to screens (starting with screen 'bbbbbb').	NO
SaaaaXObbbbbbCPcccccc	X-refernces of segement 'aaaa' to Call of P.M.S.(-CP) of screen 'bbbbbb' (starting with macro-structure 'cccccc').	NO
SaaaaXObbbbbbWccnnn	X-references of segment 'aaaa' to work areas (-W) of screen'bbbbbb' (starting with work area 'cc', line number 'nnn').	NO
SaaaaSSbn	Definition of the sub-schemas or sub-systems of segment 'aaaa' in the PACTABLE function (starting with sub-schema 'n' with 'b' = 's', or sub-system 'n' with 'b' = 'y'.	YES
SaaaaCEbbb	Call of elements/attributes of segment 'aaaa'(starting with line number 'bbb').	YES
SaaaaCEbbbGccc	General Documentation for the element/attribute called on line 'bbb' of segment 'aaaa' (starting with general documentation line number 'ccc').	YES
SaaaaDBEbbb	SQL view source for view 'aaaa' (starting with line 'bbb').	YES
SaaaaLALbbb	Level, address and length of segment 'aaaa' (starting with line 'bbb').	NO

SaaaaDEDbbb	Data element details of segment 'aaaa' (starting with line 'bbb').	NO
	If this choice is used in C2 option, the relational label replaces that of the data element.	NO
SaaaaCNbbbbbb	List of constraints of segment 'aaaa' integrity (from the block 'bbbbbb')	NO
SaaaaSTA	Statistics on segment 'aaaa'.	NO
SaaaaACT	Activity calculation on segment 'aaaa'.	NO

NOTE: After the first choice of type 'Saaaa', 'Saaaa' can be replaced with '-'.

All notations between parentheses are optional.

DATABASE BLOCKS: ON-LINE ACCESS

LISTS

CHOICE	SCREEN	UPD
LCBaaaaaa	List of database blocks by code (starting with block 'aaaaaa').	NO
LTBaabbbbbb	List of database blocks by type (starting with type 'aa' and block 'bbbbbb').	NO
LEBaaaaaaaa	List of database blocks by external name (starting with name 'aaaaaaaa').	NO

DESCRIPTION OF BLOCK 'aaaaaa'

CHOICE	SCREEN	UPD
Baaaaaa	Definition of database block 'aaaaaa'	YES
BaaaaaaGbbb	General documentation for block 'aaaaaa' (starting with line 'bbb').	YES
BaaaaaaATbbbbbb	Text assigned to block 'aaaaaa' (starting with text 'bbbbbb').	NO
BaaaaaaX	X-references of block 'aaaaaa'.	NO
BaaaaaaXBbbbbbb	X-references of block 'aaaaaa' to PSB's (starting with PSB 'bbbbbb').	NO
BaaaaaaXObbbbbb	X-references of block 'aaaaaa' to screens (starting with screen 'bbbbbb').	NO
BaaaaaaXObbbbbbCScdddd	X-references of block 'aaaaaa' to the Call of Segments of screen 'bbbbbb' (starting with category 'c' and with segment 'dddd'). Note: 'c' is equal to & for the screen-top category.	NO
BaaaaaaXObbbbbbWccddd	X-references of block 'aaaaaa' to the Work Areas of screen 'bbbbbb' (starting with work area 'cc', line number'ddd').	NO
BaaaaaaXQbbbbbb	List of entities linked to block 'aaaaaa' through user-defined relationship 'bbbbbb'.	NO
BaaaaaaXVvvvvvv	X-references of block 'aaaaaa' to volumes (starting with volume 'vvvvvv').	NO
BaaaaaaXPbbbbbb	X-references of block 'aaaaaa' to programs (starting with program 'bbbbbb').	NO
BaaaaaaXPbbbbbbWccddd	X-references of block 'aaaaaa' to Work Areas of program 'bbbbbb' (starting with work area 'cc', line number 'ddd').	NO

DESCRIPTION OF 'aaaaaa' DMSII Block

BaaaaaaDCbbb (MAJ)

Description of the 'aaaaaa' block of DMSII type (from line 'bbb').

BaaaaaaDCbbbGccc (MAJ)

Documentationof the line 'bbb' of the 'aaaaaa' Block (from the documentation line 'ccc').

NOTES: After the first choice of type 'Baaaaaa', 'Baaaaaa' can be replaced with '-'.

All notations between parentheses are optional.

6.2. BATCH

DATABASE BLOCKS: BATCH ACCESS

DEFINITION

Batch Form 'L1' is used to define a Database Block.

ACTION CODES

```
C     = Creation of a line in the library.

M     = Modification of a line.

Blank = Creation or modification of a line,  depending
        on its presence or absence in the library.

X     = Creation or modification with possible use of
        ampersands (&).

D     = Deletion of a line.

B     = Deletion of the database block and of its de-
        pendent lines.
```

DATABASE BLOCK DESCRIPTION

BATCH FORM

Batch Form 'L3' is used for the description of a CODASYL, DB2, or TANDEM Database Block.

ACTION CODES

```
.C      = Creation of a line in the library.

.M      = Modification of a line.

.Blank  = Creation or modification of a line, depending
          on its presence or absence in the library.

.X      = Creation or modification with possible use of
          ampersand (&).

.D      = Deletion of a line.

.B      = Deletion of the data base block lines starting
          from an including the indicated line number as
          well as the associated V3 lines.

.R      = End of multiple deletion following this line.
          If no R-type line appears after a B-type
          line, the deletion ends with the last line
          number of the Block.
```

DATA ELEMENT DEFINITION

Batch Form 'C' is used for the definition of a Data Element.

DATA ELEMENT DESCRIPTION

Batch Form 'E' is used for the description of a Data Element.

SEGMENT DEFINITION

Batch Form '2' is used for the definition of a Segment.

SEGMENT DESCRIPTION

Batch Form '3' is used for the description of a Segment.

ACTION CODES

The batch action codes for these entities are identical to the ones used for the Database Block entity.

NOTE CONCERNING DELETION OF A DATA ELEMENT

Deletion of a Data Element (using ACTION CODE 'D') is only possible if the Data Element is not used in screens, reports and Segments and if it has no child Data Element.

It is possible to globally delete (using ACTION CODE 'B') a Data Element and all of its uses in screens, reports or Segments.

When a multiple deletion is done on a parent Data Element, all of its child Data Elements will be deleted along with all of the uses of the parent and child Data Elements.

6.3. GENERATION AND/OR PRINTING

GENERATION AND/OR PRINTING

The generation and printing of Database Blocks is requested on-line on the Generation and Print Commands screen (CH: GP) or in batch mode on Batch Form 'Z'.

LISTS

```
LTB      Lists all database blocks of the libraries from the
         selected sub-network, sorted by type.

        .C1 OPTION: Without keywords,
        .C2 OPTION: With explicit keywords.

LCB      Identical to 'LTB' but sorted by code.

LEB      Identical to 'LTB' but sorted by external name.
```

It is possible to request a list of Database Blocks related by keyword(s). The corresponding command must be accompanied by a continuation line, on which the keywords used as selection criteria are indicated (refer to the USER'S Reference Manual). The list is sorted by code.

```
LKB      Same as 'LCB' but sorted by keyword.
         Option 'C2' cannot be used.
```

DESCRIPTION

```
DTB      Description of the database block whose code is in-
         dicated in the entity field, description of all da-
         tabase blocks if the field is not entered.
         In the latter case,  it is possible to request the
         descriptions of all blocks of a given type, by spe-
         cifying it in the printing request.
```

GENERATION OPTION

```
GCB      Generation of a Database Block whose code must be
         indicated.
         Same printing option as for DTB.
```

7. EXAMPLE

7.1. PRESENTATION

EXAMPLE PRESENTATION

The objective of this chapter is to present the different steps necessary to generate in DLL language.

The chapter contains the following parts:

. System screens used for the description of the DMSII Database (only the most significative screens are shown).

. Data description as the System generates it.

NOTE: This example is not exhaustive and does not cover all the function possibilities.

EXAMPLE 7
PRESENTATION 1

```
--------------------------------------------------------------------------------
!                                                                  *PDMCA.PDEV.HP3.8!
! DESCRIPTION OF DASDL DMSII  TDASDL TEST GENERATION DASDL                       !
!                                                                                !
! A LIN : T SET NA DATASET   OPTION     PT. COMMENT                              !
!       :   LIN IT EMB.      SET/IT     N K                                      !
!   100 : 1            DL10 S      *       MAIN FILE : COURSES                  !
!   101 : 5 PROF   DL40 DL10 C                                                   !
!   200 : 1        DL10 DL20 U      *       BOOKS                               !
!   210 : 4 LIVK   DL10 DL20 LI     *                                           !
!   300 : 1        DL10 DL30 S      *       STUDENTS                            !
!   301 : 5 SSNO   C    DL30 MFSSET                                              !
!   302 : 5 SSNO1  S    DL30 MFSSET                                              !
!   310 : 3 ETUSET DL10 DL30 IS     * 00002                                      !
!   350 : 3 COUSET      DL10 IS     *                                            !
!   400 : 1             DL40 S      *       PERSONNEL                           !
!   401 : 5 IDCOUR DL10 DL40 C                                                   !
!   402 : 5 TELEPH DL80 DL40 TELEPH                                              !
!   403 : 5 SUPER  DL40 DL40 N                                                   !
!   410 : 3 SS-U-P      DL40 IS     *                                            !
!   420 : 3 U-P-ST      DL40 IS     *                                            !
!   500 : 1             DL50 S      *       REGISTERED                          !
!   501 : 5 TELEPH DL80 DL50 TELEPH                                              !
!                                                                                !
! O: C1 CH:                                                                      !
--------------------------------------------------------------------------------
```

EXAMPLE 7
PRESENTATION 1

```
--------------------------------------------------------------------------------
!                                                                  *PDMCA.PDEV.HP3.8!
! DESCRIPTION OF DASDL DMSII  TDASDL TEST GENERATION DASDL                       !
!                                                                                !
! A LIN : T SET NA DATASET   OPTION     PT. COMMENT                              !
!       :   LIN IT EMB.      SET/IT     N K                                      !
!   510 : 3 QSET   DL50 DL60 OL     *                                           !
!   600 : 1        DL50 DL60 S      *       QUARTER                             !
!   610 : 3 CSEET  DL60 DL70 IS     *                                           !
!   700 : 1        DL60 DL70 S      * 00002 COURSES                             !
!   710 : 1 V      DL60 DL71          00001                                     !
!   720 : 1 V      DL60 DL72          00002                                     !
!   721 : 5 IDCOUR DL10 DL72 C                                                 !
!   750 : 3 MFSSET      DL50 IS     *                                          !
!   800 : 1             DL80 S      *       ADDRESS                            !
!   810 : 3 SAD         DL80 IS     *                                          !
!   820 : 3 SSAD        DL80 IS     *                                          !
!   830 : 4 STUAD       DL80 IS     * 00002                                    !
!   840 : 4 FACAD       DL80 IR     * 00002                                    !
!   850 : 4 ADMAD       DL80 IR     * 00002                                    !
!   860 : 4 FREEPA      DL40 BV     *                                          !
!   870 : 4 SEXSET      DL50 BV     *       STUDENTS OVER 21                   !
!   880 : 4 SMART       DL50 BV                                                !
!                                                                              !
! O: C1 CH:                                                                    !
--------------------------------------------------------------------------------
```

```
 ------------------------------------------------------------------------------
 !                                                             *PDMCA.PDEV.HP3.8!
 ! BLOCK DESC GENERAL DOC.         TDASDL TEST GENERATION DASDL                1!
 !                                                                              !
 ! A LIN : T COMMENT                                                        LIB !
 !   200 : P  <NOM   >ASCENDING,                                            0851!
 !   210 : P  <PRENOM>)                                                     0851!
 !   300 : Z  DUPLICATES                                                    0851!
 !   310 : Z  LOADFACTOR = 75 TABLESIZE = 12 AREAS = 100                    0851!
 !       :                                                                      !
 !       :                                                                      !
 !       :                                                                      !
 !       :                                                                      !
 !       :                                                                      !
 !       :                                                                      !
 !       :                                                                      !
 !       :                                                                      !
 !       :                                                                      !
 !       :                                                                      !
 !       :                                                                      !
 !       :                                                                      !
 !       :                                                                      !
 !       :                                                                      !
 !                                                                              !
 ! O: C1 CH:                                                                    !
 ------------------------------------------------------------------------------
```

7.2. SCREENS / BATCH TRANSACTIONS

```
XL1TDASDLTEST GENERATION DASDL                        20
XV3L1TDASDL   001V % EXEMPLE GENERATION DE DASDL
XV3L1TDASDL   010V $ SET LIST SINGLE STORE TEST
XV3L1TDASDL   900Z %%% FIN EXEMPLE
XL3TDASDL1001          DL10S     00000FICHIER PRINCIPAL : COURS
XV3L3TDASDL100100Z % FIN DE LA DESCRIPTION DE L'ENREGISTREMENT CO
XV3L3TDASDL100110Z POPULATION = 1000
XV3L3TDASDL100150Z VERIFY (DL10-NBHEU GTR 0 AND DL10-TCLASS LEQ 6
XV3L3TDASDL100151Z        AND DL10-NOPROF NEQ 0
XV3L3TDASDL100500 <IDCOUR>
XV3L3TDASDL100501P REQUIRED
XV3L3TDASDL100510 <NBINS >
XV3L3TDASDL100511G DL10-NBINS COUNT (300);
XV3L3TDASDL100520 <SALLE >
XV3L3TDASDL100521P NULL IS "NO";
XL3TDASDL1015PROF  DL40DL10C     00000
XL3TDASDL2001      DL10DL20U     00000LIVRES
XV3L3TDASDL200100Z     BUFFERS = 1 + 1 PER USER,
XV3L3TDASDL200101Z     AREAS = 10,
XV3L3TDASDL200102Z     AREASIZE = 500,
XV3L3TDASDL200103Z     POPULATION = 5,
XV3L3TDASDL200104Z     BLOCKSIZE = 5
XV3L3TDASDL200500 <TITRE >
XV3L3TDASDL200501P NULL IS BLANKS;
XL3TDASDL2104LIVK  DL10DL20LI    00000
XV3L3TDASDL210400Z DATA
XV3L3TDASDL210410Z<IDLI  >
XL3TDASDL3001      DL10DL30S     00000ETUDIANTS
XV3L3TDASDL300300Z POPULATION = 300
XL3TDASDL3015SSNO  C   DL30MFSSET00000
XL3TDASDL3025SSNO1 S   DL30MFSSET00000
XL3TDASDL3103ETUSETDL10DL30IS    00002
XV3L3TDASDL310200P<NOM   >ASCENDING,
XV3L3TDASDL310210P<PRENOM>)
XV3L3TDASDL310300Z DUPLICATES
XV3L3TDASDL310310Z LOADFACTOR = 75 TABLESIZE = 12 AREAS = 100
XL3TDASDL3503COUSET    DL10IS    00000
XV3L3TDASDL350100P<IDCOUR>DESCENDING
XV3L3TDASDL350200Z NO DUPLICATES
XL3TDASDL4001          DL40S     00000PERSONNELS
XV3L3TDASDL400100P POPULATION = 997
XV3L3TDASDL400700 <NBPER >
XV3L3TDASDL400701G DL40-NBPER COUNT (100);
XV3L3TDASDL400710 <NOMC  >
XV3L3TDASDL400711P REQUIRED
XV3L3TDASDL400720 <SALAIR>
XV3L3TDASDL400721P INITIALVALUE IS LOW-VALUE;
XV3L3TDASDL400730 <AGE   >
XV3L3TDASDL400731P NULL IS HIGH-VALUE;
XL3TDASDL4015IDCOURDL10DL40C     00000
XL3TDASDL4025TELEPHDL80DL40TELEPH00000
XL3TDASDL4035SUPER DL40DL40N     00000
XL3TDASDL4103SS-U-P    DL40IS    00000
XV3L3TDASDL410100P<SSNO  >
XV3L3TDASDL410300Z NO DUPLICATES
XL3TDASDL4203U-P-ST    DL40IS    00000
XV3L3TDASDL420100P<NOMC  >
XV3L3TDASDL420200Z DUPLICATES
XL3TDASDL5001          DL50S     00000INSCRITS
XV3L3TDASDL500100Z % FIN DESCRIPTION "INSCRITS"
XV3L3TDASDL500110Z LOCK TO MODIFY DETAILS       % OPTION PHYSIQUE
XV3L3TDASDL500120Z BLOCKSIZE = 6
XV3L3TDASDL500130Z POPULATION = 5000
XV3L3TDASDL500140Z AREAS = 100
XV3L3TDASDL500150Z KIND = DISK
XL3TDASDL5015TELEPHDL80DL50TELEPH00000
XL3TDASDL5103QSET  DL50DL60OL    00000
XV3L3TDASDL510100P<QTIER >NO DUPLICATES
XL3TDASDL6001      DL50DL60S     00000QUARTIER
XV3L3TDASDL600900Z % FIN DESCRIPTION "QUARTIER"
XL3TDASDL6103CSEET DL60DL70IS    00000
XV3L3TDASDL610100P<TYCOUR>
```

```
XV3L3TDASDL610200Z DUPLICATES
XL3TDASDL7001      DL60DL70S     00002CORSUS
XV3L3TDASDL700010V POPCORSES POPULATION (100000) OF DL70;
XV3L3TDASDL700110P POPULATION = 4, BLOCKSIZE = 30 WORDS
XL3TDASDL7101V     DL60DL71      00001
XL3TDASDL7201V     DL60DL72      00002
XL3TDASDL7215IDCOURDL10DL72C     00000
XL3TDASDL7503MFSSET    DL50IS    00000
XV3L3TDASDL750100P<SSNO  >
XV3L3TDASDL750200Z NO DUPLICATES
XL3TDASDL8001          DL80S     00000ADRESSE
XV3L3TDASDL800800ZPOPADMAD POPULATION (14) OF ADMAD
XL3TDASDL8103SAD       DL80IS    00000
XV3L3TDASDL810100Z DUPLICATES FIRST
XV3L3TDASDL810200P<ZIP   >
XL3TDASDL8203SSAD      DL80IS    00000
XV3L3TDASDL820100P<SSNO  >
XV3L3TDASDL820200Z DUPLICATES LAST
XL3TDASDL8304STUAD     DL80IS    00002
XV3L3TDASDL830010P WHERE (DL80-FACETU EQL 1)
XV3L3TDASDL830100P<ZIP   >,
XV3L3TDASDL830110P<SSNO  >)
XV3L3TDASDL830200Z DUPLICATES
XL3TDASDL8404FACAD     DL80IR    00002
XV3L3TDASDL840010P WHERE (DL80-FACETU EQL 2)
XV3L3TDASDL840100P<ZIP   >,
XV3L3TDASDL840110P<SSNO  >)
XV3L3TDASDL840200Z DUPLICATES MODULUS = 97
XL3TDASDL8504ADMAD     DL80IR    00002
XV3L3TDASDL850010P WHERE (DL80-FACETU EQL 3)
XV3L3TDASDL850100P<ZIP   >,
XV3L3TDASDL850110P<SSNO  >)
XV3L3TDASDL850200Z DUPLICATES
XL3TDASDL8604FREEPA    DL40BV    00000
XV3L3TDASDL860010P WHERE (DL40-SALAIR LSS 0 OR DL40-SALAIR EQL 0)
XL3TDASDL8704SEXSET    DL50BV    00000LES MAJEURS
XV3L3TDASDL870010P WHERE (DL50-AGE GEQ 21 AND NOT DL50-SEXE)
XL3TDASDL8804SMART     DL50BV    00000
XL3TDASDL8904DMUTIL    DL50BV    00000
```

EXAMPLE 7
DESCRIPTION OF DASDL GENERATED ELEMENTS 3

7.3. DESCRIPTION OF DASDL GENERATED ELEMENTS

```
 % DASDL GENERATION EXAMPLE
 $ SET LIST SINGLE STORE TEST
INITIALIZE;
DL10  STANDARD   DATASET
"MAIN FILE : COURSES          "
    (
    DL10-IDCOUR    GROUP
    REQUIRED
        (
        DL10-DEPART    ALPHA(2);
        DL10-NIVEAU    NUMBER(3);
        DL10-COURSN    NUMBER(4);
        );
    DL10-NOPROF    NUMBER(2);
    DL10-NBINS COUNT (300);
    DL10-SEMAIN    FIELD
        (
        DL10-LUNDI     BOOLEAN;
        DL10-MARDI     BOOLEAN;
        DL10-MERCDI    BOOLEAN;
        DL10-JEUDI     BOOLEAN;
        DL10-VENDDI    BOOLEAN;
        DL10-SAMEDI    BOOLEAN;
        );
    DL10-IMMEU     NUMBER(3);
    DL10-SALLE     ALPHA(2)
    NULL IS "NO";
    DL10-COURS     ALPHA(24);
    DL10-FLAGS     FIELD(12);
    DL10-NBHEU     NUMBER(4);
    DL10-TCLASS    NUMBER(2);
    DL10-PROF
                   IS IN  DL40  COUNTED
                   OCCURS 3 TIMES;
    DL20  UNORDERED  DATASET
    "BOOKS                              "
        (
        DL20-IDLI      NUMBER(9);
        DL20-TITRE     ALPHA(60)
        NULL IS BLANKS;
        DL20-AUTEUR    ALPHA(30);
        )
        BUFFERS = 1 + 1 PER USER,
        AREAS = 10,
        AREASIZE = 500,
        POPULATION = 5,
        BLOCKSIZE = 5
    ;
    LIVK
        SUBSET OF  DL20
        UNORDERED LIST
        DATA
        DL20-IDLI
    ;
    DL30  STANDARD   DATASET
    "STUDENTS                           "
        (
        DL30-NOM       ALPHA(15)
                       REQUIRED;
        DL30-PRENOM    ALPHA(10)
                       REQUIRED;
        DL30-SSNO
                       IS IN     MFSSET;
        DL30-SSNO1
                       IS KEY OF MFSSET;
        )
        POPULATION = 300
    ;
    ETUSET
        SET    OF  DL30
        KEY IS (
        DL30-NOM    ASCENDING,
```

```
            DL30-PRENOM )
            INDEX SEQUENTIAL
            DUPLICATES
            LOADFACTOR = 75 TABLESIZE = 12 AREAS = 100
        ;
        )
        % RECORD COURSES : END OF DESCRIPTION
        POPULATION = 1000
        VERIFY (DL10-NBHEU GTR 0 AND DL10-TCLASS LEQ 60)
        AND DL10-NOPROF NEQ 0
    ;
    COUSET
        SET    OF  DL10
        KEY IS
        DL10-IDCOUR DESCENDING
        INDEX SEQUENTIAL
        NO DUPLICATES
    ;
    DL40  STANDARD   DATASET
    "PERSONNEL                          "
        POPULATION = 997
        (
        DL40-NBPER COUNT (100);
        DL40-NOMC      GROUP
        REQUIRED
            (
            DL40-NOM       ALPHA(15);
            DL40-PRENOM    ALPHA(10);
            );
        DL40-SEXE      BOOLEAN;
        DL40-AGE       NUMBER(2)
        NULL IS HIGH-VALUE;
        DL40-SSNO      NUMBER(9)
                       REQUIRED;
        DL40-DPT       ALPHA(4);
        DL40-RANG      ALPHA(1);
        DL40-SALAIR    NUMBER(S7,2)
        INITIALVALUE IS LOW-VALUE;
        DL40-IDCOUR
                       IS IN  DL10  COUNTED
                       OCCURS 8 TIMES;
        DL40-TELEPH
                       IS IN  DL80  VERIFY ON  DL80-TELEPH;
        DL40-SUPER
                       IS IN  DL40  WITH NO PROTECTION;
        )
    ;
    SS-U-P
        SET    OF  DL40
        KEY IS
        DL40-SSNO
        INDEX SEQUENTIAL
        NO DUPLICATES
    ;
    U-P-ST
        SET    OF  DL40
        KEY IS
        DL40-NOMC
        INDEX SEQUENTIAL
        DUPLICATES
    ;
    DL50  STANDARD   DATASET
    "REGISTERED                         "
        (
        DL50-SSNO      NUMBER(9)
                       REQUIRED;
        DL50-NONOM     NUMBER(1);
        DL50-LNOM      ALPHA(30);
        DL50-ALIAS     ALPHA(30)
                       OCCURS 9;
        DL50-FNOM      ALPHA(30);
        DL50-ADRCAM    GROUP
            (
            DL50-DORTOI    ALPHA(6);
            DL50-ROOM      NUMBER(4);
            DL50-BOXE      NUMBER(4);
            DL50-POSTE     NUMBER(7);
```

```
        );
    DL50-ND         NUMBER(2);
    DL50-DEGRE      ALPHA(4)
                    OCCURS 6;
    DL50-TOTHEU     NUMBER(3);
    DL50-TOTQP      REAL(3);
    DL50-MPTGRA     NUMBER(3,2);
    DL50-MJR        NUMBER(3);
    DL50-AMJR       ALPHA(18);
    DL50-SEXE       BOOLEAN;
    DL50-AGE        NUMBER(2);
    DL50-TELEPH
                    IS IN  DL80  VERIFY ON  DL80-TELEPH;
    QSET
        SET    OF  DL60
        KEY IS
        DL60-QTIER
        NO DUPLICATES
        ORDERED LIST
    ;
    DL60  STANDARD   DATASET
    "QUARTER                              "
        (
        DL60-QTIER     ALPHA(4)
                       REQUIRED;
        DL60-QTTHRS    NUMBER(2);
        DL60-QTRQP     NUMBER(2);
        CSEET
            SET    OF  DL70
            KEY IS
            DL70-TYCOUR
            INDEX SEQUENTIAL
            DUPLICATES
        ;
            POPCORSES POPULATION (100000) OF DL70;
        DL70  STANDARD   DATASET
        "COURSES                                "
            POPULATION = 4, BLOCKSIZE = 30 WORDS
            (
            DL70-TYCOUR    NUMBER(1)
                           REQUIRED;
            DL70-CORTYP  RECORD TYPE(2);
            )
            ,
        1:
            (
            DL71-GRADE     ALPHA(2);
            DL71-IDCOUR    ALPHA(9);
            )
            ,
        2:
            (
            DL72-YR        NUMBER(2);
            DL72-COEFF     NUMBER(2);
            DL72-IDCOUR
                           IS IN  DL10  COUNTED;
            DL72-GCD       ALPHA(2);
            DL72-DIPLO     ALPHA(30);
            DL72-PPGRD     ALPHA(2);
            )
        ;
        )
        % "QUARTER" : END OF DESCRIPTION
    ;
    )
    % "REGISTERED" : END OF DESCRIPTION
    LOCK TO MODIFY DETAILS       % PHYSICAL OPTION
    BLOCKSIZE = 6
    POPULATION = 5000
    AREAS = 100
    KIND = DISK
;
MFSSET
    SET    OF  DL50
    KEY IS
    DL50-SSNO
    INDEX SEQUENTIAL
```

```
    NO DUPLICATES
;
DL80  STANDARD   DATASET
"ADDRESS                          "
    (
    DL80-FACETU    NUMBER(1);
    DL80-SSNO      NUMBER(9)
                   REQUIRED;
    DL80-NUMLNS    NUMBER(1);
    DL80-ADRLN     ALPHA(30)
                   OCCURS 9;
    DL80-ZIP       NUMBER(5)
                   REQUIRED;
    DL80-TELEPH    NUMBER(7);
    )
    ;
    POPADMAD POPULATION (14) OF ADMAD
;
SAD
    SET    OF  DL80
    KEY IS
    DL80-ZIP
    INDEX SEQUENTIAL
    DUPLICATES FIRST
;
SSAD
    SET    OF  DL80
    KEY IS
    DL80-SSNO
    INDEX SEQUENTIAL
    DUPLICATES LAST
;
STUAD
    SUBSET OF  DL80
    WHERE (DL80-FACETU EQL 1)
    KEY IS (
    DL80-ZIP    ,
    DL80-SSNO   )
    INDEX SEQUENTIAL
    DUPLICATES
;
FACAD
    SUBSET OF  DL80
    WHERE (DL80-FACETU EQL 2)
    KEY IS (
    DL80-ZIP    ,
    DL80-SSNO   )
    INDEX RANDOM
    DUPLICATES MODULUS = 97
;
ADMAD
    SUBSET OF  DL80
    WHERE (DL80-FACETU EQL 3)
    KEY IS (
    DL80-ZIP    ,
    DL80-SSNO   )
    INDEX RANDOM
    DUPLICATES
;
FREEPA
    SUBSET OF  DL40
    WHERE (DL40-SALAIR LSS 0 OR DL40-SALAIR EQL 0)
    BIT VECTOR
;
SEXSET             "STUDENTS OVER 21                  "
    SUBSET OF  DL50
    WHERE (DL50-AGE GEQ 21 AND NOT DL50-SEXE)
    BIT VECTOR
;
SMART
    SUBSET OF  DL50
    BIT VECTOR
;
DMUTIL
    SUBSET OF  DL50
    BIT VECTOR
;
```

EXAMPLE 7
DESCRIPTION OF DASDL GENERATED ELEMENTS 3

```
%%% END OF EXAMPLE
```